April 2021

Frame Recovery by Dynamic Learning and Prediction in a Video Decoder

Anonymous

[redacted] Sign Out

HOME SUBMIT PATENT SUBMIT INVENTION ALL INNOVATIONS MY INNOVATIONS MY WORK SPACE SEARCH ADMINISTRATION

Snapshot Info

Snapshot Date 12-Jun-19 **Snapshot Taken By** [redacted]

General Information

Title 2245 - Frame recovery by dynamic learning and prediction in a video decoder

Innovator [redacted]

Status [redacted]

Date Created 15-Jun-18

Submission Date 18-Jun-18

Keywords machine learning, error concealment, video decoding

Description of Invention This innovation is to use the Recurrent Neural Network (RNN), a typical machine learning model such as Long Short Term Memory (LSTM) to predict the video frames. The RNN model can be trained offline and refined in an online and dynamic fashion. The hidden states in the model can be defined as coding parameters such as frame or macroblock coding model, motion vectors, etc. If a higher frame rate is needed for a smooth video playback, this model can predict and insert frames. If a frame or more frames are missing due to various reasons, this model can place the predicted frames in the correct places. Furthermore, if the decoded frames are not IDR or I-frame or don't have large number of coded Intra blocks, the predicted frames can be used to verify and report possible quality degradation.

Categorization Fields

What was Done Before Normally when a decoder detects a missing frames or decoding errors, it issues a fast update request. While waiting for a FUR, the decoder can either freeze the display, e.g. show the last non-corrupted frame or perform some error concealments. The results of error concealment are normally very bad due to lack of correct information on motion vectors. There is no known method that can predict video frame quality at the receiver.

Advantages This innovation can learn and store the decoded information such as motion vectors, inter or intra coding modes as the hidden states and apply them in the frame predictions. As results, this approach can 1. make a better error concealment; 2. predict video image degradation before any errors are detected; 3. make smooth display by increasing video frame rate, especially for the temporal scalable video coding where the sender constantly changes the number of temporal layers due to bandwidth fluctuation.

How Easily can Use of the Invention be Detected or Discovered? With work, could test product without having to reverse engineer
Supporting Reasons:1. examine if RNN model is used in the decoder deployment; 2. examine if the hidden states are motion vectors and/or coding modes

Technology Video Processing

Project [redacted]

Invention Use Content,Streaming,Telepresence,Video Endpoints

Use in a Standard Not Applicable

Have you already or do you plan to disclose this outside of [redacted]? No

Has a prototype to test the Invention ever been created? No

Has the Invention ever been used commercially? No

Counsel

Administrative Notes

Additional Information

Electronic Documents

FILE NAME	VERSION	SIZE

Additional Contributors

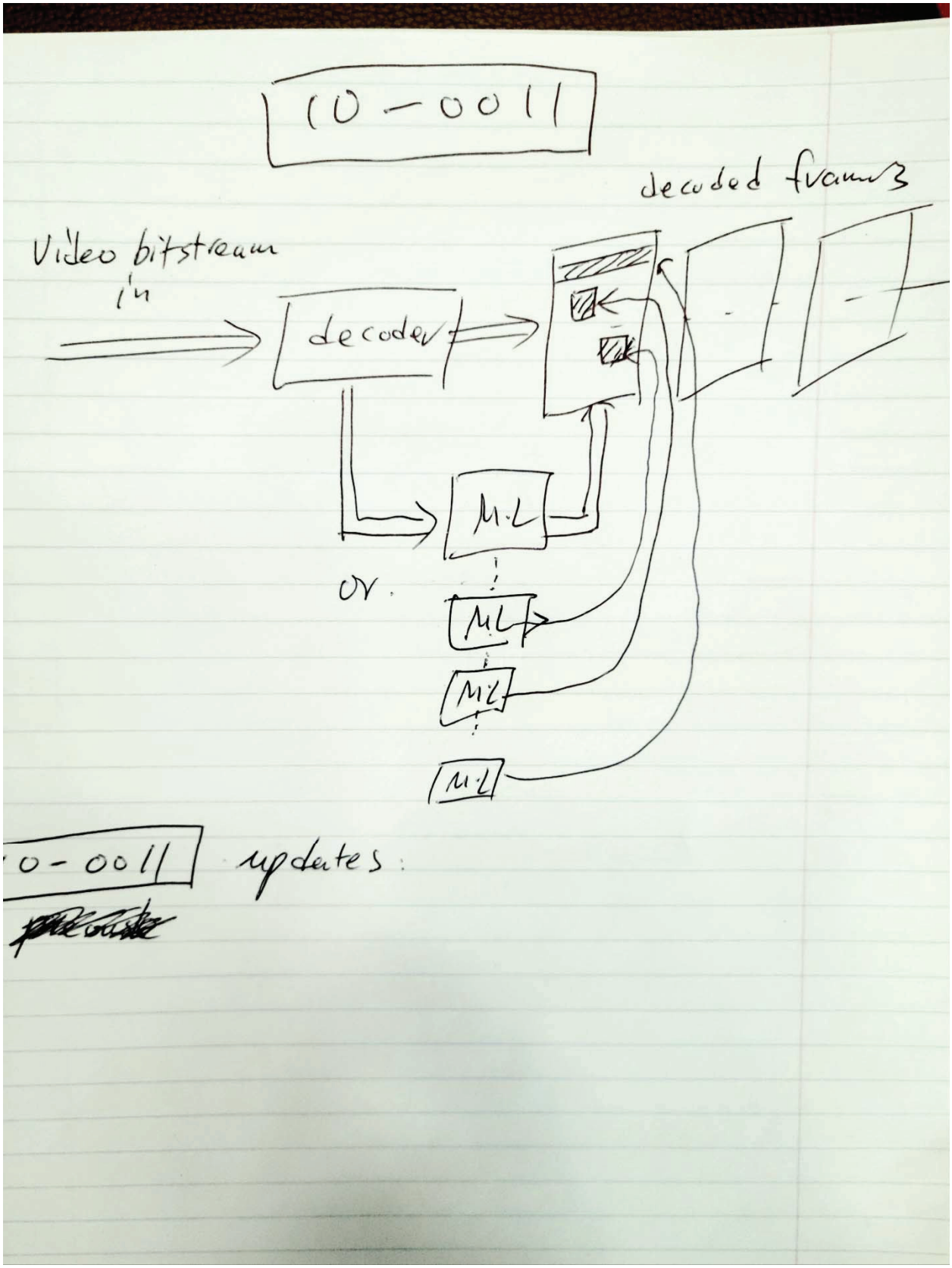
Innovator Users

NAME	EMAIL	DEPARTMENT	PHONE NUMBER

Are there contributors whose names are not available in the "Add Contributors" popup? No

Back

10-0011 Appendix



Previous research focused on Recurrent Neural Network, in fact any trained convolution neural network can perform machine learning and predicted expected parameters by aggression models.

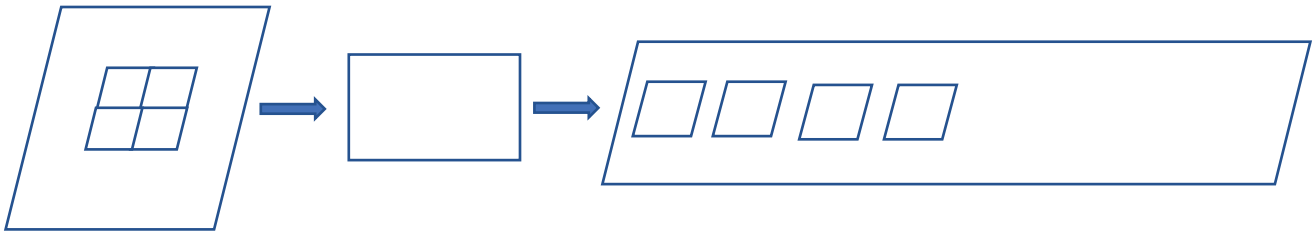
Due to video bitstream packet loss, some of coding parameters can be recovered at the decoding side, such as motion vectors, coding modes and DCT residuals. Packet lose causes missing one or more macroblocks or slices of video frames. These blocks can be estimated by its corresponding machine learning model at the same locations.

This innovation is to estimate those missing parameters from the well-trained and online updated CNN model. A big ML model can be used for estimated a whole frame and a distributed tiny ML model can be used for independent macroblocks as in drawn in the figure above.

10-0011 - Appendix Cont'd

Background Info:

1. Video encoding and packet video bit steams.
   In video compression like H.264 video frames are compressed on a basis of coding unit like 16x16 macroblock and video bit streams are generated in a form of NAL (Network Abstraction Layer) units. In general, such a bit stream consists of video packets, like SPS (Sequence Parameter Set), PPS (Picture Parameter Set), slice info and macroblock (16x16 pixels) info.



A video frame, e.g. 1920x1080 and macroblocks in the frame

A video encoder

Packet video bit stream in NAL units, SPS, PPS, Slice info, MB info …

Figure 1. Video encoding and packet video bit streams.

2. Video packet loss and video error concealment
   In IP based video communications, video bit stream packetizations can be done in different ways such as STAP, MTAP, FUP etc. defined in IETF.
   For important packet payloads like SPS and PPS that establish a video communication, if there is a packet loss down the route from an encoder to a decoder, a fast update is immediately requested.
   For other payloads, such as video packets in P or B frames, some packets may be dropped due to network congestions. As the result, error concealments can be used to estimate the missing video packets and so as to recover the missing macroblock information.

The innovation:

This innovation is to apply machine learning for a video decoding system to recover missing macroblock information in a forward-prediction mode (IPPP…) or a bi-directional prediction mode (IBBBP…)

More specifically, the machine learning engine works on the **basic coding block** like macroblock level **instead of a frame level**, such that a tine ML model is formed (the tiny size is referred to that of a frame based model):

1. The use of local regions (spatial and temporal) can greatly reduce the complexity of a ML engine on training and referencing required for a full frame

10-0011 - Appendix Cont'd

2. The use of local regions (spatial and temporal) can reduce the amount of data required for a full frame and improve the accuracy of the ML model.
3. All these **tiny ML models** run independently, a desired feature for parallel processing in devices like GPUs.
4. The tiny models are update dynamically and independently.
5. The tiny models are video codec independent.
6. The tiny models are spatial-temporal jointed learning.

The ML engine on the video decoder:

1. Temporal prediction and concealment mode for coding blocks like MBs

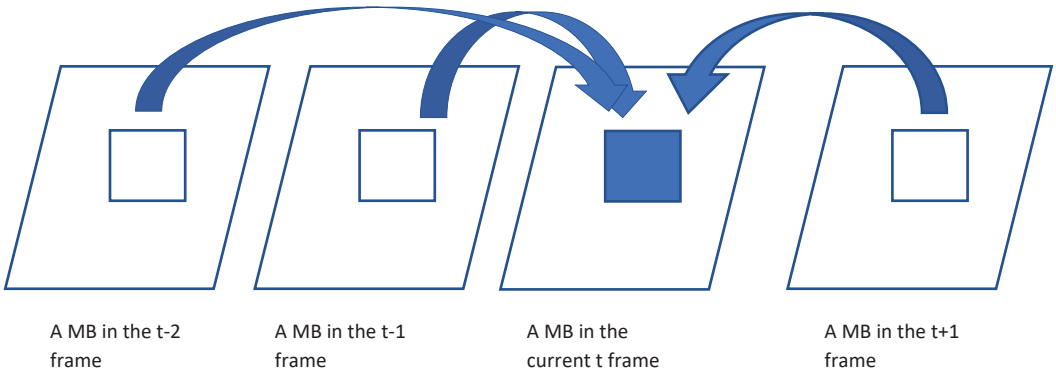
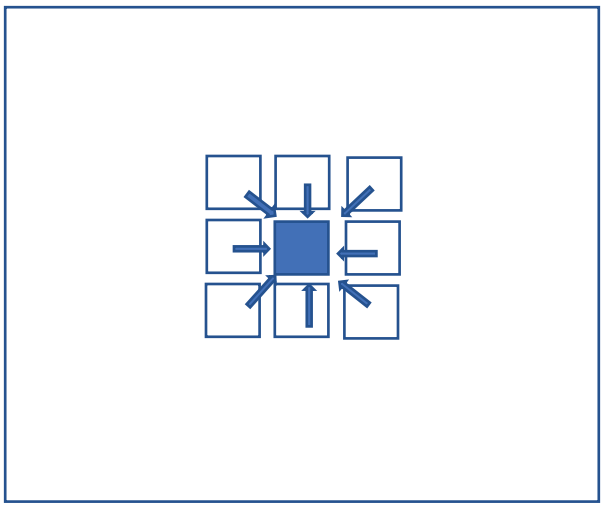


Figure 2. An illustration of a missing MB in the current frame that can be predicted from its available temporal local neighbors like a bi-directional prediction. **The concealment can be expanded to missing video slices, including multiple MBs.**

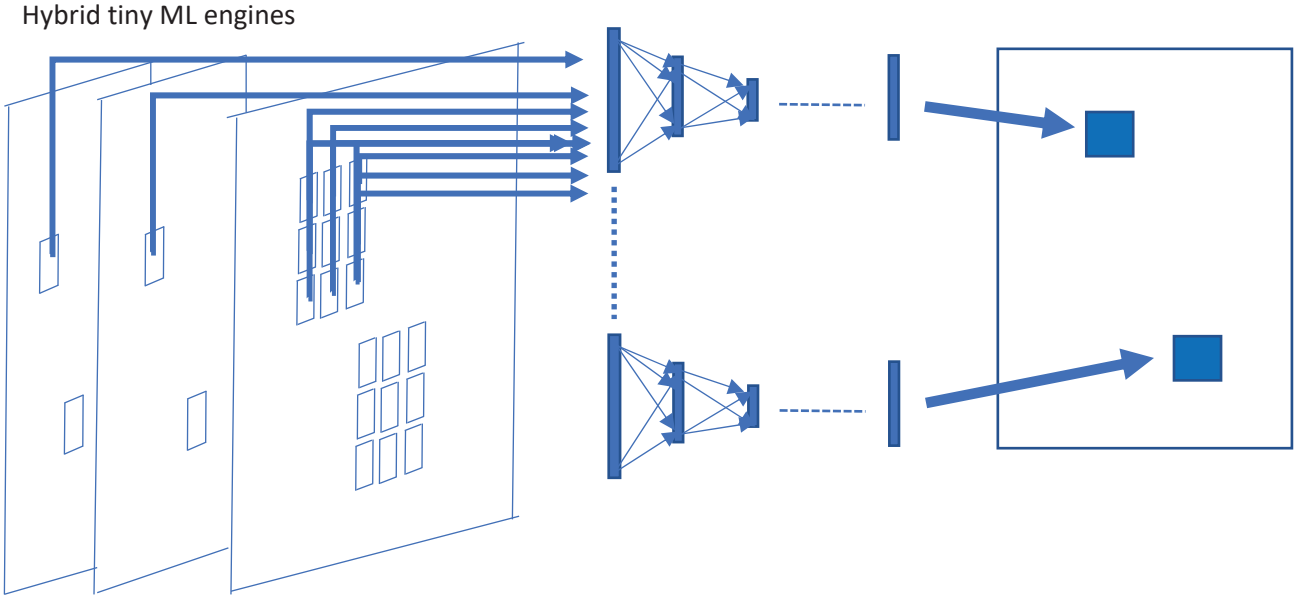
2. Spatial prediction and concealment mode



10-0011 - Appendix Cont'd

Figure 3. An illustration of a missing MB in the current frame that can be predicted from its spatial local neighbors like a 3x3 window. **The concealment can be expanded to missing video slices, including multiple MBs.**

3. Hybrid tiny ML engines



(a) Spatial and temporal available neighboring MBs

(b) A tiny ML engine for the corresponding MB

(c) the Recovery of missing MBs in a decoded frame

Figure 4. An illustration of independent and parallel tiny ML engines for the recovery of missing MBs.

The hybrid prediction of a missing MB in the tiny ML engine can be written as

$$p(x,y) = \sum\sum\big(w(i,j,x,y)f(x-i,y-j)\big) + \lambda(x,y)\sum(w(t,x,y)f(t,x,y))$$

Where *(x,y)* is the MB position in a frame, *(x-i, y-j), i ≠ j* is the spatial neighboring MB, *t* is the time instance. **The input features can be the coding mode, motion vectors, prediction mode, residuals and the pixel values of the coding blocks**. The first part of the formula is the spatial mode and the second part is the temporal mode. **The both weights can be trained and updated dynamically and parallelly.**

The weights can be stored in a memory associated with decoded frames and can be accessed by individual MB as needed.

10-0011 - Appendix Cont'd

4. An implementation of the system and method: